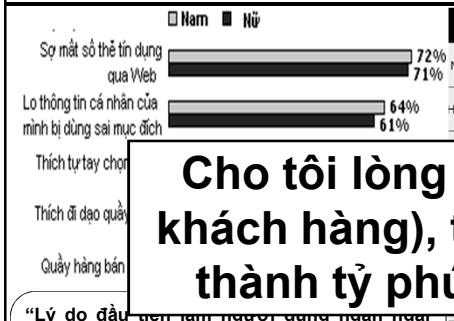
## Chương 6 An ninh trong TMĐT

1. Nguyên nhân trở ngại TMĐT phát triển
2. Vấn đề an ninh cho các hệ thống TMĐT
3. Các giao thức bảo mật
4. An ninh trong TMĐT

1

## 1. Nguyên nhân trở ngại TMĐT phát triển




| Các trở ngại triển khai TMĐT tại Việt Nam (với hệ số 4 là trở ngại lớn nhất) | |
|---|---|
| Người dân và DN chưa có thói quen | 3,23 |
| Hệ thống thanh toán điện tử còn bất cập | 3,19 |
| | 2,78 |
| | 2,64 |
| hưa | 2,45 |
| hàng | 2,45 |
| Hạ tầng CNTT và VT chưa đáp ứng yêu cầu | 2,22 |

*Nguồn: khảo sát của vụ TMĐT, bộ TM*

**Cho tôi lòng tin (của khách hàng), tôi sẽ trở thành tỷ phú (USD)**

**"Lý do đầu tiên làm người dùng ngần ngại khi sử dụng TMĐT là lo bị mất thông tin thẻ tín dụng, bí mật cá nhân bị dùng sai mục đích."**

www.ecommercetimes.com, 2003

2

## 2. Vấn đề an ninh cho các hệ thống TMĐT

- TMĐT gắn liền với giao dịch, thẻ tín dụng, séc điện tử, tiền điện tử…
- Rủi ro trong thương mại truyền thống đều xuất hiện trong TMĐT dưới hình thức tinh vi, phức tạp hơn.
- Tội phạm trong TMĐT tinh vi, phức tạp hơn
- Các hệ thống an ninh luôn tồn tại điểm yếu
- Vấn đề an ninh với việc dễ dàng sử dụng là hai mặt đối lập
- Phụ thuộc vào vấn đề an ninh của Internet, an ninh thanh toán, số lượng trang web…

3

## 2. Vấn đề an ninh cho các hệ thống TMĐT

- Một số dạng tấn công tin học trong TMĐT
  - Phần mềm độc hại (virus, trojan, worm)
  - Tin tặc
  - Gian lận thẻ tín dụng
  - Tấn công từ chối dịch vụ (DOS)
  - Phishing (kẻ giả mạo)

4

## 3. Các giao thức mật mã

- Mật mã giải quyết các vấn đề có liên quan đến bí mật, xác thực, tính toàn vẹn, và chống phủ định
- Giao thức là một chuỗi các bước, liên quan đến hai hoặc nhiều bên, được thiết kế để thực hiện một nhiệm vụ
  - "Chuỗi các bước": giao thức có một trình tự, từ đầu đến cuối
    - Mỗi bước phải được thực hiện lần lượt, và không bước nào có thể được hiện trước khi bước trước nó kết thúc

5

## 3. Các giao thức mật mã

- "Liên quan đến hai hay nhiều bên": ít nhất hai người được yêu cầu hoàn thành giao thức
  - ✓ Một người một mình không tạo nên được một giao thức. Một người một mình có thể thực hiện một loạt các bước để hoàn thành một nhiệm vụ, nhưng điều này không phải là một giao thức.
- "Được thiết kế để hoàn thành một nhiệm vụ": giao thức phải đạt được cái gì đó

6

## 3. Các giao thức mật mã

- Tất cả mọi người tham gia trong giao thức phải
  - Biết giao thức và tất cả các bước để làm theo
  - Đồng ý làm theo nó
- Giao thức phải rõ ràng
  - Mỗi bước phải được xác định rõ ràng
  - Không có cơ hội để hiểu lầm
- Giao thức phải được hoàn thành
  - Phải có một hành động cụ thể cho mọi tình huống có thể xảy ra

7

## 3. Các giao thức mật mã

- Một giao thức mật mã liên quan đến một số thuật toán mật mã, nhưng nói chung, mục tiêu của giao thức không phải là những bí mật đơn giản
- Các bên có thể muốn
  - Chia sẻ một phần bí mật để tính toán một giá trị
  - Cùng nhau tạo ra một chuỗi ngẫu nhiên
  - Thuyết phục một người khác về sự xác thực của mình
  - Hoặc đồng thời ký một hợp đồng

8

## 3. Các giao thức mật mã

- Cốt lõi của việc sử dụng mật mã học trong một giao thức là ngăn chặn hoặc phát hiện nghe lén và gian lận
  - Không nên làm nhiều hơn hoặc tìm hiểu nhiều hơn những gì được quy định trong giao thức

9

## Danh sách những người tham gia thường xuyên

- Alice: Người thứ nhất tham gia vào tất cả các giao thức
- Bob: Thứ hai tham gia trong tất cả các giao thức
- Trent: Trọng tài tin cậy

10

## Giao thức trọng tài

- Trọng tài: bên thứ ba đáng tin cậy giúp hoàn thành giao thức giữa hai hai bên không tin tưởng
- Trong thế giới thực, luật sư thường được sử dụng như các trọng tài
  - Ví dụ: Alice bán một chiếc xe cho Bob, là một người lạ. Bob muốn thanh toán bằng séc, nhưng Alice không có cách nào để biết séc là có hiệu lực.

11

## Giao thức trọng tài

- Nhờ một luật sư đáng tin cậy cho cả hai. Với sự giúp đỡ của luật sư, Alice và Bob có thể sử dụng giao thức sau đây để đảm bảo rằng không có ai gian lận
- (1) Alice trao quyền cho luật sư
- (2) Bob gửi séc cho Alice
- (3) Alice đặt cọc séc
- (4) Sau khi chờ đợi một khoảng thời gian cụ thể để séc được làm rõ ràng, luật sư trao quyền cho Bob. Nếu séc không rõ ràng trong khoảng thời gian cụ thể, Alice chứng minh với luật sư và luật sư trả trao quyền lại cho Alice.

12

## Giao thức phân xử

- Bởi vì chi phí thuê trọng tài cao, giao thức trọng tài có thể được chia thành 2 giao thức con
  - Giao thức con không có trọng tài, thực thi tại mọi thời điểm các bên muốn hoàn thành giao thức
  - Giao thức con có trọng tài, thực thi chỉ trong hoàn cảnh ngoại lệ - khi có tranh chấp

13

## Giao thức phân xử

- Ví dụ: giao thức ký kết hợp đồng có thể được chính thức hóa theo cách này
  - Giao thức con không có trọng tài (thực thi ở mọi thời điểm):
    - (1) Alice và Bob đàm phán các điều khoản của hợp đồng
    - (2) Alice ký hợp đồng
    - (3) Bob ký hợp đồng

14

## Giao thức phân xử

- Giao thức con phân xử (chỉ thực thi khi có tranh chấp):
  - (4) Alice và Bob xuất hiện trước một quan tòa
  - (5) Alice đưa ra bằng chứng của mình
  - (6) Bob trình bày bằng chứng của mình
  - (7) Quan tòa phán quyết dựa trên bằng chứng

15

## Trao đổi khóa với mã đối xứng

- Giả sử Alice và Bob muốn chia sẻ một khóa bí mật với nhau thông qua Key Distribution Center (KDC) là trọng tài trong giao thức
  - Các khóa này phải được thực hiện trước khi giao thức bắt đầu
  - (1) Alice gọi trọng tài và yêu cầu khóa phiên dùng chung để giao tiếp với Bob
  - (2) Trọng tài tạo ra một khóa phiên ngẫu nhiên, mã hóa hai bản sao của nó: một bằng khóa của Alice và một bằng khóa của Bob. Trọng tài gửi cả 2 bản copy tới cho Alice.

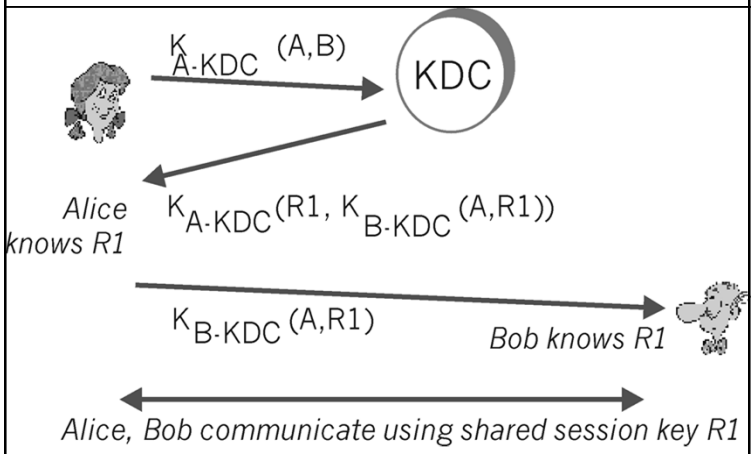
16

## Trao đổi khóa với mã đối xứng

- (3) Alice giải mã bản sao của khóa phiên
- (4) Alice gửi cho Bob bản sao của khóa phiên
- (5) Bob giải mã bản sao của khóa phiên
- (6) Cả Alice và Bob dùng khóa phiên để giao tiếp an toàn

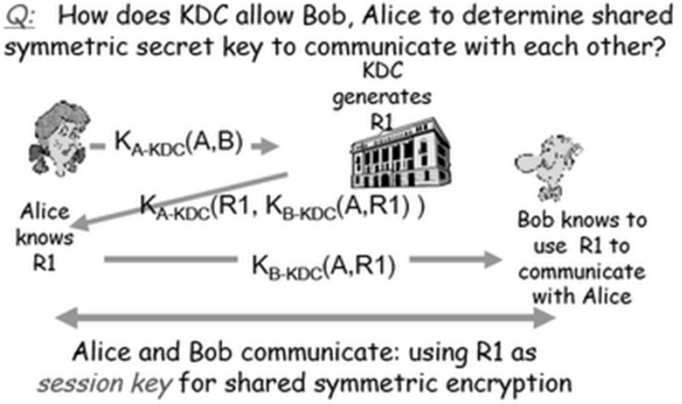
17

## Trao đổi khóa với mã đối xứng




18

## Trao đổi khóa với mã đối xứng




19

## Trao đổi khóa với mã bất đối xứng

- Alice và Bob sử dụng mật mã khóa công khai để thống nhất về khóa phiên dùng chung, và dùng khóa phiên đó để mã hóa dữ liệu
- Trong một số triển khai thực tế, cả hai khóa công khai của Alice và Bob sẽ luôn có sẵn trong CSDL

20

## Trao đổi khóa với mã bất đối xứng

- (1) Alice nhận khóa công khai của Bob từ KDC
- (2) Alice tạo ra một khóa phiên ngẫu nhiên, mã hóa nó bằng cách sử dụng khóa công khai của Bob và gửi nó đến Bob
- (3) Bob sau đó giải mã thông điệp của Alice bằng cách sử dụng khóa riêng của mình
- (4) Cả hai mã hóa các thông tin liên lạc sử dụng cùng một khóa phiên

21

## Giao thức Needham-Schroeder

- (0) Trước khi các giao dịch có thể diễn ra, mỗi người sử dụng trong hệ thống có một khóa bí mật chia sẻ với Trent.
- (1) Alice gửi một thông điệp đến Trent bao gồm tên của mình, tên Bob, và một số ngẫu nhiên: A, B, $R_A$
- (2) Trent tạo ra một khóa phiên ngẫu nhiên K, mã hóa thông điệp bao gồm khóa phiên ngẫu nhiên và tên của Alice bằng khóa bí mật của Bob. Sau đó, mã hóa giá trị ngẫu nhiên của Alice, tên của Bob, khóa, và thông điệp mã hóa bằng khóa bí mật chia sẻ với Alice, và gửi Alice mã hóa: $E_A$ ($R_A$, B, K, $E_B$($K_{22}$, A))

## Giao thức Needham-Schroeder

- (3) Alice giải mã tin nhắn và rút ra K. Alice khẳng định rằng $R_A$ là giá trị mà mình đã gửi Trent trong bước (1). Sau đó, Alice gửi Bob tin nhắn được Trent mã hóa bằng khóa của Bob: $E_B$(K, A)
- (4) Bob giải mã tin nhắn và rút ra K. Sau đó, Bob tạo ra một giá trị ngẫu nhiên, $R_B$, mã hóa tin nhắn với K và gửi nó cho Alice: $E_B$($R_B$)
- (5) Alice giải mã các tin nhắn với K, tạo ra $R_B$ - 1 và mã hóa nó với K, sau đó gửi tin nhắn cho Bob: $E_B$($R_B$ - 1)

23

## Chữ ký mù

- Đặc tính tất yếu của các giao thức chữ ký số là người ký biết những gì mình ký
- Chúng ta muốn mọi người ký các văn bản mà không bao giờ nhìn thấy nội dung
  - Bob là một công chứng viên. Alice muốn Bob ký một tài liệu, nhưng không muốn anh ta có bất kỳ ý tưởng về những gì mình ký.
    - Bob không quan tâm những gì tài liệu nói, anh ta chỉ xác nhận rằng mình có công chứng tại một thời gian nhất định. Bob sẵn sàng làm điều này.

24

## Chữ ký mù

- (1) Alice có các tài liệu và nhân bản nó bằng một giá trị ngẫu nhiên (multiple). Giá trị ngẫu nhiên này được gọi là một yếu tố làm mù.
- (2) Alice gửi tài liệu mù Bob
- 3) Bob ký tài liệu mù
- (4) Alice phân tách các yếu tố làm mù, để lại tài liệu gốc có chữ ký của Bob

25

## Lược đồ chữ ký mù

- Bước 1: A làm mù x bằng một hàm: z=Blind(x), và gửi z cho B
- Bước 2: B ký trên z bằng hàm y = Sign(z) = Sign(Blind(x)), và gửi lại y cho A.
- Bước 3: A xóa mù trên y bằng hàm. Sign(x) = UnBlind(y) = UnBlind(Sign(Blind(x)))

26

## Chữ ký mù

- Các thuộc tính của chữ ký mù hoàn chỉnh
- 1. Chữ ký của Bob lên tài liệu là hợp lệ
  - Chữ ký là một minh chứng rằng Bob đã ký các tài liệu
  - Nó sẽ thuyết phục Bob rằng anh ta đã ký các tài liệu nếu nó đã từng được hiển thị cho anh ta
  - Nó cũng có tất cả các thuộc tính khác của chữ ký số
- 2. Bob không thể đánh đồng các văn bản được ký kết với các hành vi ký kết các tài liệu
  - Ngay cả nếu Bob giữ hồ sơ của tất cả các chữ ký mù, Bob không thể xác định mình đã ký tài liệu nào

27

## 4. An ninh trong TMĐT

- Bảo mật giao dịch thanh toán
- Bảo mật tiền số
- Bảo mật séc điện tử

28

## 4.1. Bảo mật giao dịch thanh toán

- Giao dịch thanh toán điện tử là sự thực thi các giao thức mà theo đó một khoản tiền được lấy từ người trả tiền và chuyển cho người nhận
- Trong một giao dịch thanh toán, chúng ta thường phân biệt giữa các thông tin đặt hàng (hàng hóa, dịch vụ phải trả) và tài liệu thanh toán (ví dụ, số thẻ tín dụng)
- Từ góc độ an ninh, hai loại thông tin này cần thiết phải được xử lý đặc biệt

29

## 4.1. Bảo mật giao dịch thanh toán

1. Nặc danh người dùng và không theo dõi thanh toán
2. Nặc danh người thanh toán
3. Không theo dõi giao dịch thanh toán
4. Bảo mật dữ liệu giao dịch thanh toán
5. Thông điệp chống phủ định giao dịch thanh toán

30

## 4.1.1. Nặc danh người dùng và không theo dõi

- Đặc tính nặc danh người dùng và không theo dõi thanh toán có thể được cung cấp riêng biệt
- Dịch vụ an ninh nặc danh người dùng "tinh khiết" sẽ bảo vệ chống lại tiết lộ xác thực người dùng
- Dịch vụ an ninh không theo dõi thanh toán "tinh khiết" sẽ bảo vệ chống lại tiết lộ địa điểm của thông điệp gốc
- Giải pháp: Sử dụng chuỗi hỗn hợp Mixes

31

## Chuỗi hỗn hợp Mixes

- Ý tưởng
  - Thông điệp được gửi từ A, B, và C (đại diện cho khách hàng có yêu cầu nặc danh) tới hỗn hợp và từ hỗn hợp tới X, Y, Z (đại diện cho các người bán hoặc ngân hàng "tò mò" về thông tin xác thực khách hàng)
  - Thông điệp được mã hóa với khóa công khai của hỗn hợp, $E_M$. Nếu khách hàng A muốn gửi thông điệp cho người bán Y, A gửi tới hỗn hợp cấu trúc sau:
    - ⇨ A → Mix: $E_M(Y, E_Y(Y, \text{Message}))$
  - Bây giờ, hỗn hợp có thể giải mã và gửi kết quả cho Y:
    - ⇨ Mix → Y: $E_Y(Y, \text{Message})$

32

## Chuỗi hỗn hợp Mixes

- Chỉ có Y mới đọc được thông điệp khi nó được mã bằng khóa công khai của Y, $E_Y$
- Nếu A muốn Y phản hồi, A có thể hàm chứa địa chỉ phản hồi nặc danh trong thông điệp gửi tới Y: Mix, $E_M(A)$
  - ⇨ A sẽ tạo một khóa Kx là khóa phiên dùng chung (chỉ dùng 1 lần) với Y. Và một khóa S1 là khóa ngẫu nhiên dùng để niêm phong
  - ⇨ A sẽ gửi Mix thông điệp:
  - ⇨ A → Mix: $E_M(Y, E_Y(Y, Message, Mix, E_M(A,S1),Kx))$
  - ⇨ hỗn hợp Mix có thể giải mã và gửi kết quả cho Y
  - ⇨ Mix → Y: $E_Y(Y, Message, Mix, E_M(A,S1),Kx)$
  - ⇨ Y → Mix: $E_M(A,S1), Kx(Response)$
  - ⇨ Mix → A: $S1(Kx(Response))$

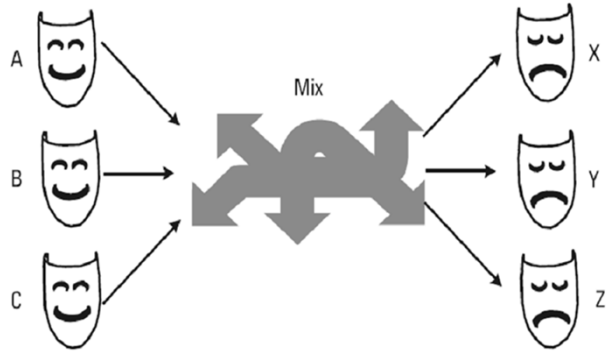
33

## Chuỗi hỗn hợp Mixes

- Theo cách này, thông điệp phản hồi được gửi tới mix, chỉ mix biết ai là người gửi
- Hạn chế
  - Hỗn hợp phải hoàn toàn đáng tin cậy

34

## Chuỗi hỗn hợp Mixes

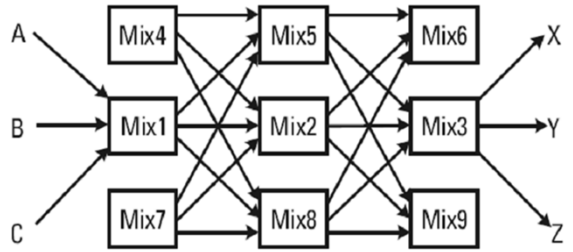



Chaum's mix.

35

## Chuỗi hỗn hợp Mixes

- Giải quyết vấn đề tin cậy của hỗn hợp, sử dụng 1 chuỗi các hỗn hợp (mạng)




Chain of mixes.

36

## Chuỗi hỗn hợp Mixes

- Nếu A muốn gửi 1 thông điệp nặc danh, không bị theo dõi tới Y, A sử dụng giao thức sau:
  - A → Mix1: $E_1(\text{Mix2}, E_2(\text{Mix3}, E_3(\text{Y}, \text{Message})))$
  - Mix1 → Mix2: $E_2(\text{Mix3}, E_3(\text{Y}, \text{Message}))$
  - Mix2 → Mix3: $E_3(\text{Y}, \text{Message})$
  - Mix3 → Y: Message
  - $E_{\text{Recipient}}(\text{Next recipient}, E_{\text{Next recipient}}(\ldots))$

37

## 4.1.2. Sự nặc danh người thanh toán

- Người trả tiền sử dụng bút danh thay vì sự nhận dạng của mình
- Nếu một bên muốn chắc chắn rằng hai giao dịch thanh toán khác nhau thực hiện bởi cùng một người không thể được liên kết với nhau, khi đó đặc tính không theo dõi giao dịch thanh toán phải được cung cấp
- Giải pháp: Sử dụng bút danh

38

## Bút danh

- Hệ thống First Virtual
  - Thông điệp ủy quyền giữa FV và người bán trước khi chuyển hàng cần phải được bảo vệ để ngăn chặn việc chuyển số lượng hàng lớn tới khách hàng gian lận
  - Khách hàng nhận VPIN (Virtual Personal Identification Number), là 1 chuỗi kí tự chữ và số hoạt động như là bút danh cho thẻ tín dụng, có thể được gửi qua email
  - Nếu VPIN bị đánh cắp, khách hàng không có thẩm quyền cũng không thể sử dụng vì tất cả các giao dịch sẽ được xác nhận bằng email trước khi trừ tiền trong thẻ tín dụng

39

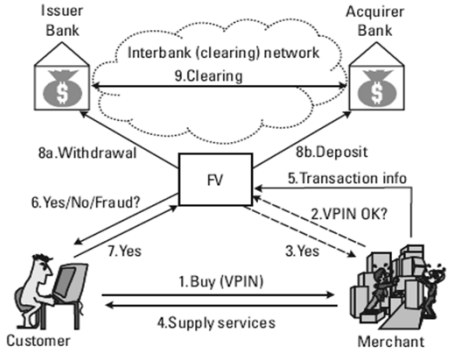
## Bút danh

- Hệ thống First Virtual
  - Nếu khách hàng cố gắng sử dụng VPIN mà không được ủy quyền, FV sẽ được thông báo về VPIN bị đánh cắp khi khách hàng phản hồi “fraud” (có sự gian lận) cho yêu cầu của FV để xác nhận mua bán
  - Trong trường hợp này VPIN sẽ bị hủy bỏ ngay lập tức
  - Cơ chế này đảm bảo tính bí mật của lệnh thanh toán đối với người bán và kẻ nghe trộm tiềm năng

40

## Bút danh

- Hệ thống First Virtual




First Virtual's (FV) payment system.

41

## Bút danh

- Giao dịch của hệ thống FV
  - (1) Khách hàng gửi đơn hàng tới người bán cùng với VPIN
  - (2) Người bán gửi yêu cầu cấp phép VPIN tới nhà cung cấp dịch vụ thanh toán FV
  - (3) Nếu VPIN là hợp lệ
  - (4) Người bán cung cấp dịch vụ cho khách hàng
  - (5) Người bán gửi thông tin giao dịch cho FV
  - (6) FV hỏi khách hàng xem đã sẵn sàng thanh toán cho các dịch vụ (qua e-mail) (Khách hàng có thể từ chối thanh toán khi dịch vụ đã được chuyển đi nhưng không đáp ứng mong đợi của mình)

42

## Bút danh

- Giao dịch của hệ thống FV
  - (7) Nếu khách hàng muốn thanh toán, trả lời "Yes"
  - (8a) Số lượng thanh toán được trừ trong tài khoản của khách hàng
  - (8b) Gửi vào tài khoản người bán
  - (9) Giao dịch bù trừ giữa các ngân hàng

43

## 4.1.3. Không theo dõi giao dịch thanh toán

- Hashsum ngẫu nhiên trong thanh toán iKP
- Hashsum ngẫu nhiên trong SET

44

## Hashsum ngẫu nhiên trong thanh toán iKP

- Khi khởi tạo 1 giao dịch thanh toán, khách hàng chọn 1 giá trị ngẫu nhiên $R_C$ và tạo ra giá trị bút danh dùng 1 lần $ID_C$ theo cách sau:

  $$ID_C = h_k(R_C, BAN)$$

  - BAN là số tài khoản ngân hàng của khách hàng
  - $h_k(.)$ là đụng độ hash 1-chiều, không tiết lộ thông tin BAN khi $R_C$ được chọn ngẫu nhiên
  - Người bán nhận $ID_C$ (không phải BAN), không thể tính ra BAN

45

## Hashsum ngẫu nhiên trong SET

- Người bán nhận 1 giá trị hashsum từ lệnh thanh toán
- Lệnh thanh toán chứa các thông tin:
  - Số tài khoản chính, PAN (ví dụ, số thẻ tín dụng)
  - Ngày hết hạn
  - Khóa chia sẻ bí mật giữa chủ thẻ, cổng thanh toán và chứng thực ủy quyền của chủ thẻ, PANSecret
  - Giá trị ngẫu nhiên nonce ngăn chặn tấn công từ điển, EXNonce
  - Khi nonce là khác nhau với mỗi giao dịch, người bán không thể liên kết 2 giao dịch với nhau ngay cả khi dùng chung PAN

46

## 4.1.4. Bảo mật dữ liệu giao dịch thanh toán

- Hàm số ngẫu nhiên giả
- Chữ ký kép (Dual signature)

47

## Hàm số ngẫu nhiên giả

- Thanh toán iKP là 1 họ các giao thức thanh toán (i = 1, 2, 3) được phát triển bởi IBM
- Hỗ trợ giao dịch thẻ tín dụng, cùng với CyberCash, Secure Transaction Set Technology và các giao thức thanh tóan điện tử an toàn là hình thức nguyên thủy quan trọng nhất của SET

48

## Hàm số ngẫu nhiên giả

- Cơ chế 1KP cung cấp tính bảo mật của các thông tin với cổng thanh toán, người bán, sự nặc danh khách hàng
  - Khi khởi tạo giao dịch, khách hàng chọn 1 giá trị ngẫu nhiên $R_C$ và tạo bút danh dùng 1 lần $ID_C$:

    $ID_C = h_k(R_C, BAN)$
  - BAN là số tài khoản ngân hàng của khách hàng
  - $h_k(.)$ là đụng độ hash k-chiều, không tiết lộ thông tin BAN khi $R_C$ được chọn ngẫu nhiên
  - Người bán nhận $ID_C$ (không phải BAN), không thể tính ra BAN

49

## Hàm số ngẫu nhiên giả

- Bảo mật thông tin tương ứng với ngân hàng thanh toán được thực hiện theo cách tương tự
  - Khách hàng chọn giá trị ngẫu nhiên $SALT_C$ khác nhau cho mỗi giao dịch, gửi cho người bán ở dạng dữ liệu rõ
  - Dùng hàm hash như ở trên, người bán gửi 1 mô tả của thông tin (DESC) cho ngân hàng thanh toán:

    $h_k(SALT_C, DESC)$
  - Ngân hàng thanh toán có thể nhìn thấy giá trị hashsum nhưng không đủ thông tin để tìm ra DESC

50

## Hàm số ngẫu nhiên giả

- Nếu số lượng DESC không nhiều, ngân hàng thanh toán có thể tính ra tất cả các giá trị của hashsum với $SALT_C$ cho trước và lấy thông tin
- Ngân hàng thanh toán có thể được tin tưởng trong một vài phạm vi
- Để truyền lệnh thanh toán tới ngân hàng thanh toán mà người bán không thể đọc được, iKP sử dụng khóa công khai

51

## Hàm số ngẫu nhiên giả

- Khách hàng mã hóa thông điệp, gồm:
  - Giá của những hàng hóa đã đặt
  - Lệnh thanh toán (số thẻ tín dụng, mã PIN)
  - $h_k(SALT_C, DESC)$ được băm cùng với dữ liệu giao dịch
  - Giá trị ngẫu nhiên $R_C$ để tạo bút danh dùng 1 lần cùng với khóa công khai của ngân hàng thanh toán
- Thông điệp mã hóa này được gửi cho người bán và chuyển tiếp tới ngân hàng thanh toán

52

## Hàm số ngẫu nhiên giả

- Khách hàng phải có chứng thực khóa công khai của ngân hàng thanh toán được phát hành bởi tổ chức chứng thực ủy quyền tin cậy
- Chỉ có ngân hàng thanh toán mới giải mã được thông điệp, qua $R_C$ xác thực độ chính xác của $ID_C$
- Kết nối giữa lệnh thanh toán và thông tin đặt hàng được thiết lập thông qua giá trị $h_k(SALT_C, DESC)$ và tất cả các bên đều biết
- Sự kết hợp 2 giá trị này là duy nhất cho mỗi giao dịch

53

## Chữ ký kép (Dual signature)

- SET là một tiêu chuẩn mở cho giao dịch thẻ tín dụng an toàn trên mạng
- Khởi động bởi Visa và MasterCard năm 1996, dùng công nghệ mã hóa RSA
- Để bảo vệ số thẻ tín dụng (lệnh thanh toán của khách hàng) từ việc nghe trộm hay những người bán không trung thực, SET sử dụng chữ ký kép

54

## Chữ ký kép (Dual signature)

- Kịch bản thanh toán
  - PI – lệnh thanh toán (payment instruction)
  - OI – các thông tin đặt hàng
  - M – người bán
  - P – payment gateway
  - Mục tiêu: Người bán M không thể đọc thông tin lệnh thanh toán, cổng thanh toán P không thể đọc thông tin đặt hàng
  - Thực hiện: Khách hàng thực hiện chữ ký kép lên yêu cầu thanh toán, tức là, C kí lên PI và OI dự định gửi cho P và M, sử dụng hàm mã hóa hash h(.) và khóa bí mật $D_C$ từ thuật toán khóa công khai

55

## Chữ ký kép (Dual signature)

- Khách hàng tính $DS = D_C(h(h(PI),h(OI)))$
  - Giả sử M chỉ biết OI, P chỉ biết PI, thì chỉ nhận được từng phần bí mật của hashsum
  - M nhận: OI, h(PI), DS
  - P nhận: PI, h(OI), DS
  - Cả 2 có thể xác thực DS
  - Nếu P đồng ý, lệnh thanh toán là chính xác, cấp quyền là khả thi, P kí lên PI, nếu M đồng ý, ký lên OI

56

## Chữ ký kép (Dual signature)

- Trong SET, h(PI) và h(OI) là các giá trị hashsum SHA-1
- C gửi PI tới M theo dạng mã hóa (thuật toán mã hóa đối xứng với khóa ngẫu nhiên bí mật K)
- Khóa bí mật này được mã hóa và gửi cùng khóa công khai của P, $E_P$, vì vậy chỉ P có thể đọc
- C → M: OI, h(PI), DS, $E_P$(K), $E_K$(P, PI, h(OI))
- M chuyển tiếp tất cả các thành phần của thông điệp (ngoại trừ OI) tới P trong 1 yêu cầu cấp phép

57

## 4.1.5. Chống phủ định giao dịch thanh toán

- Chống phủ định sẽ ngăn chặn việc từ chối nguồn gốc của tài liệu hoặc phủ nhận bằng chứng bàn giao
- Giải pháp: Chữ ký số

58

## Chữ ký số

- Để giải thích các vấn đề chống phủ định, ta sử dụng mô hình 3KP
  - Acquirer đại diện cho cổng thanh toán và ngân hàng thanh toán
  - Giả định các thông tin đặt hàng (hàng hóa, dịch vụ, giá cả, hình thức giao hàng) đã được thương lượng trước thông báo (yêu cầu) thanh toán
  - Thông báo thanh toán này là duy nhất để xác thực các giao dịch thanh toán
  - Người trả tiền gửi người nhận thông báo thanh toán gồm lệnh thanh toán và công cụ thanh toán xác định

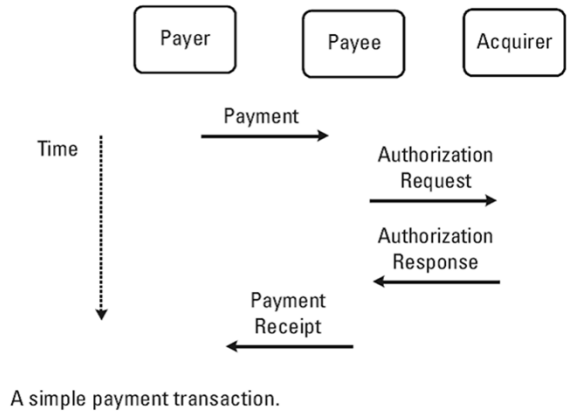
59

## Chữ ký số

- Ví dụ
  - Thẻ tín dụng có các thông tin: Ngân hàng phát hành, Số thẻ, Ngày hết hạn (thời gian hiệu lực)
  - Người nhận tiền muốn xác thực thẻ tín dụng có thể được dùng để tính toán, sẽ gửi thông báo yêu cầu ủy quyền (**authorization request**) tới ngân hàng thanh toán
  - Thông điệp trả lời ủy quyền (**authorization response**) chứa kết quả ủy quyền
  - Nếu kết quả là chắc chắn, người nhận sẽ gửi xác nhận thanh toán (**payment receipt**) cho người trả và gửi hàng hóa dịch vụ

60

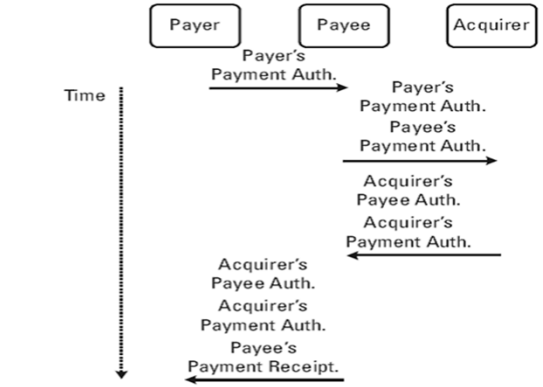
## Chữ ký số




A simple payment transaction.

61

## Chữ ký số




Nonrepudiation messages.

62

## Chữ ký số

- Vấn đề chống phủ định và ủy quyền
  - Thông điệp ủy quyền được gửi bởi 3 thành phần, mỗi thành phần có 1 cặp khóa công khai, mỗi khóa được chứng thực bởi 1 tổ chức chứng thực đáng tin cậy
  - Người nhận cần 1 bằng chứng (không thể từ chối) rằng người trả đã đồng ý trả 1 khoản tiền nhất định
  - Bằng chứng này được chứa trong thông điệp **Payer's Payment Authorization,** đảm bảo chống phủ định ủy quyền thanh toán của người trả

63

## Chữ ký số

- Vấn đề chống phủ định và ủy quyền
  - Ngân hàng thanh toán và ngân hàng phát hành cần bằng chứng **Payer's Payment Authorization** để thu hồi số tiền bán hàng từ tài khoản của người trả, ghi có tài khoản người nhận, được ký bằng khóa bí mật của người trả
  - Ngân hàng thanh toán và ngân hàng phát hành cần bằng chứng chống phủ định ủy quyền thanh tóan của người nhận, đó là chức năng của **Payee's Payment Authorization**, đảm bảo chống phủ định ủy quyền thanh toán, được ký bằng khóa bí mật của người nhận

64

## Chữ ký số

- Vấn đề chống phủ định và ủy quyền
  - Người nhận tiền hỏi Acquirer thông điệp **Acquirer's Payment Authorization** để chứng minh Acquirer đã đồng ý với giao dịch thanh tóan, được khóa bằng khóa bí mật của Acquirer
  - **Acquirer's payee auth** chứng minh rằng người nhận thanh toán được ủy quyền để nhận các khoản thanh toán
  - Giấy chứng nhận gửi cho người nhận được chuyển tiếp cho người trả, người nhận không thể từ chối là người trả đã trả cho những đơn hàng đã đặt
  - Biên lai thường được ký bởi người nhận

65

## 4.2. Bảo mật tiền số

1. Không theo dõi giao dịch thanh toán
2. Chống double spending
3. Chống làm giả xu
4. Chống đánh cắp xu

66

## 4.2.1. Không theo dõi giao dịch thanh toán

- Khi khách hàng rút tiền truyền thống từ máy ATM hoặc tài khoản ngân hàng, chuỗi series numbers trên tiền thường không được ghi lại
- Các giao dịch thanh toán không thể liên kết tới 1 khách hàng cụ thể
- Tiền số cũng có số serial numbers và đôi khi được biểu diễn bởi các số duy nhất thỏa mãn các điều kiện cụ thể

67

## 4.2.1. Không theo dõi giao dịch thanh toán

- Serial numbers tồn tại dưới dạng số rất dễ tạo ra bản ghi lưu lại thông tin khách hàng
- Vì vậy, nó rất đơn giản để thực hiện theo dõi giao dịch thanh toán điện tử của khách hàng bằng cách lần theo serial numbers
- Giải pháp
  - Chữ ký mù
  - Trao đổi xu

68

## Chữ ký mù

- D. Chaum đề xuất nhằm che giấu sự liên kết giữa các đồng xu được phát hành với thông tin xác thực khách hàng
- Cung cấp sự nặc danh cho người thanh toán và không theo dõi giao dịch thanh toán dựa trên chữ ký mù
- Người ký không nhìn thấy những gì mình ký

69

## Chữ ký mù

- Kịch bản (dựa trên thuật toán RSA)
  - d là khóa bí mật của người gửi, e và n là khóa công khai
  - Tham số k được gọi là nhân tố làm mù, được chọn bởi message provider
  - Provider làm mù thông điệp M:

$$M' = Mk^e \bmod n$$

  - Người ký thực hiện chữ ký mù:

$$S' = (M')^d \bmod n = kM^d \bmod n$$

  - Provider loại bỏ nhân tố làm mù

$$S = S'/k = M^d \bmod n$$

70

## Chữ ký mù

- Người ký thường muốn kiểm tra nếu thông điệp M (tức là, tiền giấy hay tiền số) nếu nó là hợp lệ
- Provider chuẩn bị n thông điệp và làm mù cùng với nhân tố làm mù khác nhau
- Người ký chọn n-1 thông điệp ngẫu nhiên và hỏi provider để gửi nhân tố mù tương ứng
- Người ký kiểm tra n-1 thông điệp, nếu đúng, ký lên thông điệp còn lại
- Đồng xu điện tử được làm mù theo cách này chỉ được sử dụng trong hệ thống thanh toán online, để kiểm tra double spending, phải được kiểm tra trong CSDL trung tâm

71

## Trao đổi xu

- Cơ chế nặc danh người dùng và nặc danh thanh toán dựa trên các bên thứ ba đáng tin cậy
- Sử dụng mạng các máy chủ tiền để trao đổi cơ sở xác thực xu cho những xu nặc danh, sau khi khẳng định tính hợp lệ và kiểm tra double spending
- Kiểu nặc danh này yếu hơn chữ ký mù
  - Với chữ ký mù, không thể xác định được danh tính người dùng
  - Với máy chủ tiền, nếu các bên có âm mưu, gồm cả máy chủ tiền trong giao dịch, có thể xác định ai đã sử dụng tiền

72

## 4.2.2. Chống double spending

- Tiền số có thể được sao chép một cách dễ dàng và tùy tiện, được thực thiện bởi bất cứ ai vì nó là dữ liệu điện tử đơn giản
- Người trả tiền có 1 đồng xu có giá trị hợp lệ, có thể cố gắng chi tiêu nhiều hơn 1 lần
- Giải pháp
  - Nặc danh có điều kiện bằng cắt và chọn (cut-and-choose)
  - Người bảo vệ

73

## Nặc danh có điều kiện bằng cắt và chọn

- Được kích hoạt cho những khách hàng không trung thực
  - Khách hàng trung thực không cố gắng tiêu xu nhiều hơn 1 lần và vẫn còn nặc danh
  - Khách hàng không trung thực là những người cố gắng tiêu xu 2 lần, danh tính bị tiết lộ

74

## Nặc danh có điều kiện bằng cắt và chọn

- Cơ chế chia cắt bí mật
- Ý tưởng: chia 1 thông điệp M thành các mẩu tin và do đó tất cả các mẩu tin phải được sắp xếp cùng nhau để tái tạo lại M (trong mô hình chia cắt bí mật tổng quan, chỉ cần 1 tập con các mẩu tin là đủ)
- Tìm $M_1$ và $M_2$ sao cho $M = M_1 \oplus M_2$
- Thực hiện: chọn $M_1$ ngẫu nhiên, cùng độ dài M và tính $M_2$ theo $M_2 = M \oplus M_1$

75

## Nặc danh có điều kiện bằng cắt và chọn

- Trong tiền số, mỗi đồng xu được gán 1 chuỗi số và N cặp mã hóa khác nhau ($I_1$, $I_2$) (tức là, được mã với khóa khác nhau) để thông tin xác thực khách hàng có thể được tiết lộ $I = I_1 \oplus I_2$
- Khi khách hàng trả tiền, người bán yêu cầu khách giải mã hoặc $I_1$ hoặc $I_2$ từ mỗi cặp (chọn ngẫu nhiên)

76

## Nặc danh có điều kiện bằng cắt và chọn

- Xác minh xem kết quả giải mã là hợp lệ nếu áp dụng thuật toán mã khóa công khai
- Nếu khách hàng cố gắng tiêu xu 1 lần nữa, với N đủ lớn (N=100), ít nhất 1 phần của I giống với 1 phần của I đã được tiết lộ ở thời điểm tiêu lần đầu (từ cùng 1 cặp) sẽ được tiết lộ

77

## Người bảo vệ

- Tập cơ chế phức tạp bảo vệ chống lại double spending trong hệ thống thanh toán off-line
- Cơ chế tương tự được sử dụng wallet
- Ý tưởng: Bên phát hành là tổ chức ngân hàng phát triển tiền điện tử
- Ví (wallet) chứa ví (purse), được tin tưởng bởi người trả tiền, và một người bảo vệ được tin cậy bởi bên phát hành

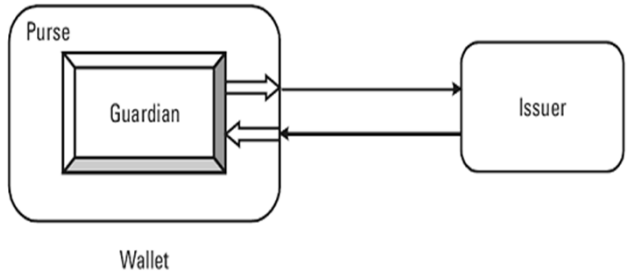
78

## Người bảo vệ

- Người bảo vệ: Là chip vi xử lý đặt cố định trong ví hoặc trên 1 thẻ thông minh
  - Bảo vệ lợi ích bên phát hành trong giao dịch thanh toán off-line
  - Là thiết bị chống trộm hoặc chống giả mạo
- Ví có dạng máy tính cầm tay nhỏ có nguồn cung cấp, bàn phím, màn hình riêng
  - Bảo vệ lợi ích của người trả tiền (nặc danh và không thể theo dõi)
  - Xác thực người bảo vệ, người bảo vệ giao tiếp ra bên ngoài thông qua purse

79

## Người bảo vệ




Electronic wallet with guardian.

80

## Chữ ký của người bảo vệ

- Người trả tiền rút tiền điện tử từ 1 tài khoản tiền xu và nạp vào ví
  - "một phần" của mỗi đồng xu được đưa vào ví
  - "một phần" khác gửi tới người bảo vệ
- Người sử dụng chi tiêu tiền trong 1 giao dịch thanh toán, người bảo vệ phải đồng ý
  - Cả hai phần của đồng xu phải được kết hợp để có được 1 đồng xu có thể được chấp nhận
  - Kết hợp 2 phần của xu được sử dụng 1 loại chữ ký số đặc biệt

81

## Chữ ký của người bảo vệ

- Tham số công khai giống như trong DSA
  - p là số nguyên tố đủ lớn
  - q là số nguyên tố đủ lớn
  - g là bộ sinh module p của q, tức là $g = h^{p-1/q} \bmod p > 1$ $(1<h<p-1)$
  - Nhóm tạo bởi g tạo nên $G_q$, giả sử người bảo vệ là bên ký, khóa gồm 2 số:
  - Số nguyên ngẫu nhiên x, $0 < x < q$ (private key)

$$h = g^x \bmod p \text{ (public key)}$$

82

## Chữ ký của người bảo vệ

- Purse muốn nhận 1 chữ ký mù từ người bảo vệ lên thông điệp $m \in G_q$
- Thông điệp có thể biểu diễn 1 đồng xu
- Chữ ký gồm: $z = m^x \bmod p$
- Chứng minh rằng: $\log_g h = \log_m z$
- Tương đương với khóa bí mật x của người bảo vệ

83

## Chữ ký của người bảo vệ

- Với m và z đã cho, giao thức sau người chứng minh (guardian) có thể chứng minh với người xác thực (purse) là biết x:
  - 1. Prover → Verifier: $a = g^s \bmod p, b = m^s \bmod p$
  - 2. Verifier → Prover: $r = \text{random}, 0 \le r < q$ challenge $c$
  - 3. Prover → Verifier: $c = \text{random}, 0 \le c < q$ response $r = (s + cx) \bmod q$

84

## Chữ ký của người bảo vệ

- 1. Verifier sẽ kiểm tra xem các biểu thức sau:

$$y^r = ah^c \bmod p$$

$$m^r = bz^c \bmod p$$

- 2. Nếu chữ ký là hợp lệ:
- 3. Chữ ký thực lên m được xác định $c = h(m, z, a, b)$

$$y^r = ah^c \bmod p$$

$$m^r = bz^c \bmod p$$

85

## Chữ ký của người bảo vệ

- H(.) là hàm băm 1 chiều
- Người bảo vệ chọn s không mất phí
- Purse ngăn chặn bằng cách xác định a và b
- Thay vì s, $s_0 + s_1$ được sử dụng, $s_0$ chọn bởi purse, $s_1$ chọn bởi người bảo vệ, các giá trị sau đây được sử dụng:

$$a = y^{s_0} h^{s_1} \bmod p$$

$$b = m^{s_0} z^{s_1} \bmod p$$

$$r = (s_0 + s_1 + cx) \bmod q$$

86

## Chữ ký bên phát hành

- Chữ ký trên đồng xu mà purse nhận từ ngân hàng phát hành phải được làm mù
  - 1. Verifier → Signer $m_0 = m^t \bmod p$, t là ngẫu nhiên, $0 < t < q$
  - 2. Signer → Verifier $a_0 = g^s \bmod p$, $b_0 = m_0^s \bmod p$, s là ngẫu nhiên, $0 < s < q$
  - 3. Verifier → Signer $c_0 = c/u \bmod q$, u là ngẫu nhiên, $0 \le u < q$
  - 4. Signer → Verifier $r_0 = (s + c_0 x) \bmod q$

87

## 4.2.3. Chống làm giả xu

- Khó làm giả tiền truyền thống
- Giấy phải đặc biệt, đắt hoặc khó sản xuất các đặc tính vật lý (in ấn)
- Số series phải hợp lệ
- Nếu là giả thì dễ kiểm tra
- Giải pháp: chi phí sản xuất xu đắt

88

## Chi phí sản xuất xu đắt

- Chi phí sản xuất những đồng xu giá trị thấp là đắt
- Nếu cần thiết để thiết lập 1 kênh đầu tư sản xuất xu (giả mạo), những xu giả mạo sẽ không thể thanh toán hết phí đầu tư
- Sản xuất nhiều xu sẽ rẻ hơn sản xuất 1 vài đồng xu
- Hệ thống thanh toán MicroMint

89

## Chi phí sản xuất xu đắt

- Sử dụng hàm hash
- Tạo đụng độ hash k-chiều ($x_1$, $x_2$, K, $x_k$), sao cho
  - $h(x_1) = h(x_2) = K = h(x_k)$
  - $h(.)$ là hàm mã hóa hash ánh xạ m-bit đầu vào ($x_i$, i = 1, K, k) sang n-bit đầu ra (hashsum)
  - Xác thực xu thực hiện bằng cách kiểm tra các giá trị x phân biệt cùng sản xuất ra hashsum giống nhau
  - Khoảng giá trị của x cần kiểm tra để nhận được đụng độ k-chiều đầu tiên (xác suất 50%)
  - Lặp c lần, $c^k$ đụng độ k-chiều được tìm thấy

90

## 4.2.4. Chống ăn cắp xu

- Một cách trực quan để bảo vệ xu khỏi bị đánh cắp thông qua nghe trộm là sử dụng mã hóa
  - Xu thường có giá trị danh nghĩa khá thấp (nhỏ hơn 1 đồng Euro)
  - Không hiệu quả khi sử dụng mã hóa
- Giải pháp: Tùy chỉnh xu

91

## Đặc trưng khách hàng và nặc danh xu

- Xu có thể được tùy chỉnh để sử dụng chỉ với khách hàng cụ thể trong giai đoạn nhất định
- Cơ chế duy trì tính nặc danh, bảo vệ chống double spending và đảm bảo khách hàng nhận biên lai hay tiền thừa hợp lệ
- Giao thức gồm 4 bước


NetCash protocol with customized coins.

92

## Đặc trưng khách hàng và nặc danh xu

- Trong Step 1 A gửi coins tới CS (currency server) để nhận về 1 đồng coin triplet
- Step1: $E_{CS}(coins, K_{AN1}, E_B, t_B, t_A)$
  - Thông điệp được mã bởi khóa công khai $E_{CS}$ của máy chủ
  - $K_{AN1}$ là khóa phiên đối xứng được sử dụng bởi CS để mã hóa bộ triplet
- Step2: $<C_B, C_A, C_X>$
  - Mỗi xu trong triplet có cùng chuỗi serial numbers và giá trị
  - B có thể sử dụng xu $C_B$ trước thời điểm $t_B$. Nếu B muốn dùng coin trong giao dịch với CS, phải chứng minh biết khóa bí mật $D_B$, khi khóa công khai $E_B$ được nhúng vào $C_B$.

93

## Đặc trưng khách hàng và nặc danh xu

- Nếu A quyết định tiêu xu với B, A gửi thông điệp tới B cho biết dịch vụ đang sử dụng (ServiceID)
- Step 3: $E_B(C_B, K_{AN2}, K_{ses}, ServiceID)$
  - B giữ lại khóa phiên $K_{ses}$
  - Tại thời điểm dịch vụ được cung cấp, B xác thực rằng A biết $K_{ses}$, B phải chuyển đổi xu khi nó có hiệu lực (trước thời điểm $t_B$)
  - Giả sử B phản hồi thông điệp trong bước 3 cùng 1 biên lai có kí tên chứa thông tin giao dịch (Amount, TransactionID) và time stamp (TS), mã với khóa phiên đối xứng $K_{AN2}$

94

## Đặc trưng khách hàng và nặc danh xu

- Step 4: $K_{AN2}(D_B(Amount, TransactionID, TS))$
  - Nếu B không gửi A biên lai, A yêu cầu CS kiểm tra xem B đã sử dụng xu. Nếu B sử dụng xu, CS phát hành 1 biên lai đã kí tên tới A xác định giá trị xu và khóa của B. Nếu B chưa dùng xu, A nhận 1 khoản tiền trong thời gian $C_A$ có hiệu lực.
  - A có thể sử dụng xu $C_A$ sau $t_B$ trước $t_A$. A quyết định không tiêu xu với B ($C_B$) nhưng sử dụng trong giao dịch với CS ($C_A$), A phải chứng minh biết khóa riêng $D_A$, khi khóa công khai $E_A$ được nhúng trong $C_A$
  - Cuối cùng, $C_X$ được dùng nếu A không tiêu xu với B

95

## Đặc trưng khách hàng và nặc danh xu

NetCash Coin Triplet

| Coin | May be spent by | Validity period |
|---|---|---|
| $C_B$ | B | Before $t_B$ |
| $C_A$ | A | From $t_B$ to $t_A$ |
| $C_X$ | Anybody | After $t_A$ |

96

## 4.3. Bảo mật séc điện tử

- Giải pháp: Chuyển giao ủy quyền thanh toán
  - Proxies
    - Kerberos
    - Restricted Proxy
    - Cascaded Proxies

97

## Chuyển giao ủy quyền thanh toán

- Sự ủy quyền thanh toán được chuyển từ người trả sang người nhận kèm theo 1 số hạn chế nhất định
- Cơ chế chữ ký điện tử lên séc điện tử dựa trên những proxies hạn chế được sử dụng để cài đặt cho hệ thống NetCheque

98

## Proxies

- Hệ thống NetCheque được phát triển bởi Information Sciences Institute of the University of Southern California
- Ban đầu là 1 dịch vụ phân tán để bảo trì hạn mức cho tài nguyên hệ thống phân tán
- Hỗ trợ mô hình thanh toán dựa trên credit-debit
- Trong mô hình credit, khoản phí được gửi tới 1 tài khoản và khách hàng thanh toán lượng tiền yêu cầu cho dịch vụ thanh toán sau

99

## Proxies

- Trong mô hình debit, tài khoản được ghi nợ khi séc (giao dịch ghi nợ) được xử lý
- Cơ chế mô tả trong phần này áp dụng cho mô hình debit
- Séc NetCheque là tài liệu điện tử, gồm:
  - Payer's name, Payer's account identifier (number) & bank name
  - Payee's name, The amount of money, The issue date
  - Payer's electronic signature, Payer's electronic endorsement (chứng thực điện tử của người trả)

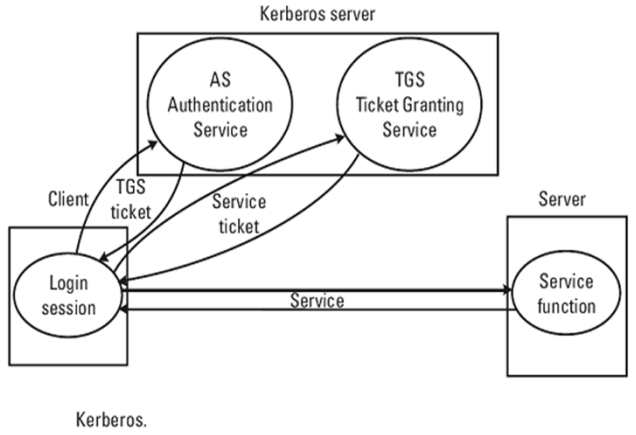
100

## Kerberos

- Proxy là thẻ cho phép thực hiện với quyền và độ ưu tiên mà một bên cho phép với proxy
- Restricted proxy là proxy kèm theo những hạn chế
- Trong ví dụ séc điện tử, sự hạn chế là các người nhận tiền (designated customer), số lượng tiền được thanh toán và ngày phát hành
- NetCheque proxies dựa trên Kerberos, phát triển bởi MIT năm 1986, ban đầu là hệ thống chứng thực phân tán

101

## Kerberos




Kerberos.

102

## Kerberos

- Client có "mong muốn" sử dụng dịch vụ S trong hệ thống phân tán, nhận service ticket từ TGS
  - Chứng thực bản thân với AS (authenticate service)
  - Nếu thành công, C nhận TGS ticket và khóa phiên $K_{C\text{-}TGS}$ để yêu cầu service ticket từ TGS

    $\{C, TGS, t_1, t_2, K_{C\text{-}TGS}\}K_{TGS}, \{K_{C\text{-}TGS}, n_1\}K_C$
  - $t_1$, $t_2$ là mốc bắt đầu và kết thúc của giai đoạn xác thực ticket
  - $n_1$, $n_2$ là các giá trị nonces (xâu ngẫu nhiên)
  - $K_{TGS}$ là khóa bí mật của TGS, $K_C$ là khóa bí mật của client

103

## Kerberos

- Client yêu cầu một service ticket
  - TGS gửi client service ticket và khóa phiên $K_{C\text{-}S}$ để yêu cầu dịch vụ

    $\{C, S, t_1, t_2, K_{C\text{-}S}\}K_S, \{K_{C\text{-}S}, n_2\}K_{C\text{-}TGS}$
  - $K_S$ là khóa bí mật của server
  - Nếu service ticket là hợp lệ, client được phép dùng dịch vụ
  - Tất cả các khóa (ngoại trừ $K_{C\text{-}S}$) được biết bởi Kerberos server, mỗi server đều phải chia sẻ khóa bí mật với các server khác

104

## Restricted Proxies

- Hệ thống Kerberos TGS ticket trên thực tế là một restricted proxy
- Hạn chế ở đây là khoảng thời gian ($t_1$, $t_2$) trong đó ticket là hợp lệ
- Dạng tổng quan của sự ủy restricted proxy:

  {restrictions, $K_{proxy}$}$K_{grantor}$, {$K_{proxy}$, nonce}$K_{grantee}$
- Grantor là thành phần đại diện cho proxy cho phép truy cập (tức là, TGS)

105

## Restricted Proxies

- Grantee là thành phần được chỉ định để thay thế grantor (tức là dịch vụ khách hàng). Restriction được biểu diễn bởi dữ liệu séc:

  {<check>, $K_{proxy}$}$K_{payer}$, {$K_{proxy}$, nonce}$K_{payee}$

106

## Cascaded proxies

- Thực tế, người trả tiền và người nhận tiền không cần phải có tài khoản tại cùng một ngân hàng
- Khi đó, séc sẽ được bù trừ thông qua nhiều hệ thống Accounting server trong NetCheque system
- Khách hàng tạo ra 1 Kerberos ticket được dùng để chứng thực khách hàng với Accounting server
- Được đặt trong thành phần chữ ký của séc và gửi cho người bán (bước 1)

  Proxy 1:{<check>, $K_{proxy1}$}$K_{customer}$,{$K_{proxy1}$, $n_1$}$K_{merchant}$

107

## Cascaded proxies

- Người bán tạo ra 1 chứng thực xác thực séc dưới tên của người nhận để đặt cọc tiền chỉ gửi vào tài khoản của người nhận (bước 2)
- Người bán gửi chứng thực cùng thông điệp gốc của khách tới Accounting Server đầu tiên (AS1)

  Proxy 2:{deposit<check> to $AS_1$, $K_{proxy2}$}$K_{proxy1}$, {$K_{proxy2,}$ $n_2$}$K_{AS1}$
- $AS_1$ chia sẻ khóa bí mật $K_{merchant}$ với người bán, có thể nhận $K_{proxy1}$ từ Proxy 1 và dùng mã hóa ticket từ Proxy 2

108

## Cascaded proxies

- Cuối cùng, $AS_1$ tạo 1 chứng thực cho tờ séc dưới tên của người người nhận để đặt tiền vào tài khoản tương ứng với $AS_1$ tại $AS_2$

  Proxy 3:{deposit <check> to $AS_2$, $K_{proxy3}$}$K_{proxy2}$, {$K_{proxy3}$, $n_3$}$K_{AS2}$
- Cả 3 cascaded proxies được gửi tới Accounting server của khách hàng $AS_2$
- Server xác thực cascaded proxies cùng ticket trong Proxy1, trao đổi khóa bí mật $K_{customer}$ với khách hàng
- $AS_2$ nhận $K_{proxy1}$ dùng để giải mã ticket trong Proxy 2

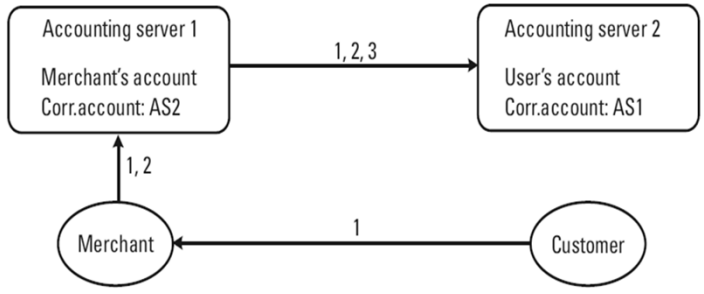
109

## Cascaded proxies

- Thông qua $K_{proxy2}$ từ Proxy 2, $AS_2$ giải mã ticket từ Proxy 3
- Ticket này sẽ cho biết séc nên được đặt cọc vào tài khoản tương ứng của $AS_1$ hay không

110

## Cascaded proxies




An accounting hierarchy.

111